Practitioner's Docket No. NAI1P004/00.006.01 *PATENT*

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re application of: D. Melchione

Application No.: 09/585,811 Group No.: 2132
Filed: 05/31/2000 Examiner: Gurshman
For: SYSTEM, METHOD AND COMPUTER PROGRAM PRODUCT FOR PROCESS-BASED SELECTION OF VIRUS DETECTION ACTIONS

Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

ATTENTION: Board of Patent Appeals and Interferences

APPELLANT'S BRIEF (37 C.F.R. § 41.37)

This brief is in furtherance of the Notice of Appeal, filed in this case on March 03, 2005.

The fees required under § 1.17, and any required petition for extension of time for filing this brief and fees therefor, are dealt with in the accompanying TRANSMITTAL OF APPEAL BRIEF.

This brief contains these items under the following headings, and in the order set forth below (37 C.F.R. § 41.37(c)(i)):

I REAL PARTY IN INTEREST
II RELATED APPEALS AND INTERFERENCES
III STATUS OF CLAIMS
IV STATUS OF AMENDMENTS
V SUMMARY OF CLAIMED SUBJECT MATTER
VI ISSUES
VII ARGUMENTS
VIII APPENDIX OF CLAIMS INVOLVED IN THE APPEAL

IX APPENDIX LISTING ANY EVIDENCE RELIED ON BY THE APPELLANT IN THE APPEAL

The final page of this brief bears the practitioner's signature.

I REAL PARTY IN INTEREST (37 C.F.R. § 41.37(c)(1)(i))

The real party in interest in this appeal is Networks Associates Technology, Inc.

II RELATED APPEALS AND INTERFERENCES (37 C.F.R. § 41.37(c)(1)(ii))

With respect to other prior or pending appeals, interferences, or related judicial proceedings that will directly affect, or be directly affected by, or have a bearing on the Board's decision in the pending appeal, there are no other such appeals, interferences, or related judicial proceedings.

Since no such proceedings exist, no Related Proceedings Appendix is appended hereto.

III STATUS OF CLAIMS (37 C.F.R. § 41.37(c) (1)(iii))

A. TOTAL NUMBER OF CLAIMS IN APPLICATION

Claims in the application are: 1, 3-7, 9-13, 15-18, and 24-29

B. STATUS OF ALL THE CLAIMS IN APPLICATION

1. Claims withdrawn from consideration: none
2. Claims pending: 1, 3-7, 9-13, 15-18, and 24-29
3. Claims allowed: None
4. Claims rejected: 1, 3-7, 9-13, 15-18, and 24-29

C. CLAIMS ON APPEAL

The claims on appeal are: 1, 3-7, 9-13, 15-18, and 24-29

See additional status information in the Appendix of Claims.

IV STATUS OF AMENDMENTS (37 C.F.R. § 41.37(c)(1)(iv))

As to the status of any amendment filed subsequent to final rejection, there are no such amendments after final.

V SUMMARY OF CLAIMED SUBJECT MATTER (37 C.F.R. § 41.37(c)(1)(v))

With respect to a summary of Claim 1 et al., a technique is provided for on-access computer virus scanning of files in an efficient manner. See Figure 2. As set forth in operation 204 of Figure 2 and the accompanying description on page 8 of the originally filed specification, a process for accessing files is identified. Further, virus detection actions are selected based at least in part on the process. Note operation 206 of Figure 2 and the accompanying description on page 8 of the originally filed specification. As set forth in operation 208 of Figure 2 and the accompanying description on page 8 of the originally filed specification, the virus detection actions are performed on the files. As set forth on page 11 of the originally filed specification, the process may identified from a plurality of processes each carried out by an executable file, where the processes includes at least one process initiated by an application program selected from the group consisting of a network browser application and a word processor application, for tailoring the virus detection actions when the application program attempts to access the files.

With respect to a summary of Claim 24 et al., a technique is provided for computer virus scanning of files in an efficient manner. As set forth in Figure 4 and the accompanying description of the originally filed specification, the extension of the file being accessed is determined so that virus detection actions may be performed on the file based on whether the extension is defined by the user. As set forth on page 11 of the originally filed specification, at least a portion of the extensions relates to a plurality of processes each carried out by an executable file, where the processes includes at least one process initiated by an application program selected from the group consisting of a network browser application and a word processor application, for tailoring the virus detection actions when the application program attempts to access the files.

With respect to a summary of Claim 29 et al., a technique is provided for on-access computer virus scanning of files in an efficient manner. See Figures 2 and 4. As set forth in operation 204 of Figure 2 and the accompanying description on page 8 of the originally filed specification, a process for accessing files is identified. Further, virus detection actions are selected based at least in part on the process. Note operation 206 of Figure 2 and the accompanying description on page 8 of the originally filed specification. As set forth in operation 208 of Figure 2 and the accompanying description on page 8 of the originally filed specification, the virus detection actions are performed on the files. As set forth on page 11 of the originally filed specification, the process may identified from a plurality of processes each carried out by an executable file, where the processes initiated by application program-related executable files include FindFast.exe, WinWord.exe, and Explorer.exe, for tailoring the virus detection actions when attempts are made to access the files. Further, the virus detection actions are selected by determining a category associated with the process, and selecting a set of virus detection actions based on the determined category. See Figure 4 and the accompanying description of the originally filed specification. Still yet, the process is identified by inspecting a name of the process, a path of the process, a file signature associated with the process, a version of the process, a manufacturer of the process, a function being called during the process, an owner of the process, a name of an executable file associated with the process, a method in which files are being accessed by the process, type(s) of shared libraries used by the identified process, and a user of the process. See page 10 et al. of the originally filed specification.

VI ISSUES (37 C.F.R. § 41.37(c)(1)(vi))

Following, under each issue listed, is a concise statement setting forth the corresponding ground of rejection.

Issue # 1: The Examiner has rejected Claims 1, 3-7, 9-13, 15-18, and 24-29 under 35 U.S.C. 103(a) as being unpatentable over Ranger (USPN 6,393,568) in view of Ji (USPN 5,623,600).

VII ARGUMENTS (37 C.F.R. § 41.37(c)(1)(vii))

The claims of the groups noted below do not stand or fall together. In the present section, appellant explains why the claims of each group are believed to be separately patentable.

Issue #1:

The Examiner has rejected Claims 1, 3-7, 9-13, 15-18, and 24-29 under 35 U.S.C. 103(a) as being unpatentable over Ranger (USPN 6,393,568) in view of Ji (USPN 5,623,600).

Group #1: Claims 1, 6-7, 12-13, 18, 26-28

With respect to the present grouping, the Examiner continues to rely on the following excerpt from Ranger to make a prior art showing of appellant's claimed "identifying a process for accessing files" (see Claims 1, 7, 13, and 28 et al.).

> "A computer based encryption and decryption system is disclosed which provides content analysis through a content inspection mechanism, such as detection of a computer virus using a virus detection mechanism, based on determining whether digital input information is encrypted." (see col. 2, lines 25-28)

Moreover, in the Examiner's Final Office Action mailed 12/06/04, the Examiner argues that "Ranger teaches the content analysis through a content inspection mechanism." Appellant respectfully disagrees with this assertion. Content analysis in no way meets the specificity of appellant's claimed identification of a process for accessing files. It appears that the Examiner is simply not taking into account the full weight of appellant's claims. Appellant does not merely analyze the content of data. Instead, appellant teaches and claims identification of a process that is used to access files, wherein virus detection actions are selected based at least in part on such file-accessing process.

Again, the foregoing excerpt merely suggests identifying whether a file is encrypted or not, so that the file can be decrypted prior to scanning. There is simply no disclosure, teaching or even suggestion of any sort of identification of "a process for accessing files," as claimed. Appellant respectfully asserts that the determination of whether a file is encrypted or not (i.e. the state of file) in no way suggests the identification of a process that is accessing the file. Only appellant teaches and claims a technique for tailoring virus detection actions based on processes that access files.

The Examiner continues by arguing that "Ji teaches identifying the process by determining whether the file is [an] executable module." Appellant respectfully disagrees with such line of reasoning. In particular, merely determining a file extension of a file (to determine whether it is executable) simply does not meet appellant's claimed identification of a process for accessing files. Any number of processes may be used to access different files regardless of file type, and Ji makes absolutely no suggestion of identifying such processes for the purpose claimed.

In response to appellant's previously-filed amendments/arguments, the Examiner now relies on the following excerpts from Ji to make a prior art showing of appellant's claimed "wherein the process is identified from a plurality of processes each carried out by an executable file, the processes including at least one process initiated by an application program selected from the group consisting of a network browser application and a word processor application, for tailoring the virus detection actions when the application program attempts to access the files" (see this or similar, but not necessarily identical, language in each of the independent claims).

> "A system for detecting and eliminating viruses on a computer network includes a File Transfer Protocol (FTP) proxy server, for controlling the transfer of files and a Simple Mail Transfer Protocol (SMTP) proxy server for controlling the transfer of mail messages through the system. The FTP proxy server and SMTP proxy server run

> concurrently with the normal operation of the system and operate in a manner such that viruses transmitted to or from the network in files and messages are detected before transfer into or from the system. The FTP proxy server and SMTP proxy server scan all incoming and outgoing files and messages, respectively before transfer for viruses and then transfer the files and messages, only if they do not contain any viruses. A method for processing a file before transmission into or from the network includes the steps of: receiving the data transfer command and file name; transferring the file to a system node; performing virus detection on the file; determining whether the file contains any viruses; transferring the file from the system to a recipient node if the file does not contain a virus; and deleting the file if the file contains a virus." (see Abstract)
>
> "This step is preferably performed by checking the extension of the file name. For example, .txt, .bmd, .pcx and .gif extension files indicate that the file is not likely to contain viruses while .exe, .zip, and .com extension files are of the type that often contain viruses. If the file to be transferred is not of a type that can contain viruses, then the method continues in step 612." (see col. 7, lines 35-40)

Further, the Examiner argues that "Ji teaches determining whether the file to be transferred is of a type that can contain viruses. This step is performed by checking the extension of the file name. For example, .txt, .bmd, .pcx. and gif extension files indicate that the file is not likely to contain viruses while .exe, .zip, and .com extension files are of the type that often contain viruses."

Whether this is true or not, Ji (in combination with Ranger) simply fails to meet appellant's claims. Specifically, reviewing file extensions simply does not rise to the level of specificity of appellant's claims. Again, any number of processes can access different files of different extensions. In other words, merely reviewing a file extension does not identify a process that is accessing the file. Moreover, Ji merely makes a blanket assertion that a word processor may reside on the computer. This,

in no way, suggests that it is determined when such word processor (or network browser) is accessing a file, so that virus detection actions may be tailored in response to such accessing.

Only appellant teaches and claims a technique that is capable of identifying an application program (i.e. a network browser application or a word processor application) that is attempting to access a file, and tailoring the virus detection actions in view of the access attempt by such specific application program.

In the Examiner's latest Response to Arguments (see Advisory Action mailed 02/15/05), with respect to the claimed "identifying a process for accessing files," the Examiner argues that "finding out if information is encrypted allows to track the decryption process (i.e. file accessing process), which is associated with the virus." Again, appellant respectfully disagrees. First, "finding out if information is encrypted" does not allow any sort of "track[ing of] the decryption process." Moreover, as argued hereinabove, determining whether information is encrypted simply does not equate to identification of a process that accesses files.

To establish a *prima facie* case of obviousness, three basic criteria must be met. First, there must be some suggestion or motivation, either in the references themselves or in the knowledge generally available to one of ordinary skill in the art, to modify the reference or to combine reference teachings. Second, there must be a reasonable expectation of success. Finally, the prior art reference (or references when combined) must teach or suggest all the claim limitations. The teaching or suggestion to make the claimed combination and the reasonable expectation of success must both be found in the prior art and not based on appellant's disclosure. *In re Vaeck*,947 F.2d 488, 20 USPQ2d 1438 (Fed.Cir.1991).

Appellant respectfully asserts that at least the first and third element of the *prima facie* case of obviousness has not been met. For example, with respect to the third element of the *prima facie* case of obvious, such element have not been met since

the prior art references, when combined, fail to teach or suggest all of the claim limitations, as noted above, and it would thus not have been obvious to one of ordinary skill in the art to modify the content analyzer of Ranger by adding the functionality for checking whether the process for accessing the file is carried out by an executable file within an application as taught by Ji, as asserted by the Examiner.

Group #2: Claims 3, 9, and 15

Regarding the present grouping, it is noted that the Examiner has still not even attempted to make a specific prior art showing of the subject matter of Claim 3 et al. See the subject matter below, which is simply non-existent in the prior art relied upon.

> "wherein the virus detection actions are selected by determining a category associated with the process, and selecting a set of virus detection actions based on the determined category" (see Claim 3 et al.)

Appellant respectfully asserts that at least the third element of the *prima facie* case of obviousness has not been met, since the prior art references, when combined, fail to teach or suggest all of the claim limitations, as noted above.

Group #3: Claims 4, 10, and 16

With respect to the present grouping, it is noted that the Examiner has still not even attempted to make a specific prior art showing of the subject matter of Claim 4 et al. See the subject matter below, which is simply non-existent in the prior art relied upon.

> "identifying the files being accessed, and selecting the virus detection actions based at least in part on the identity of the files" (see Claim 4 et al.)

Appellant respectfully asserts that at least the third element of the *prima facie* case of obviousness has not been met, since the prior art references, when combined, fail to teach or suggest all of the claim limitations, as noted above.

Group #4: Claims 5, 11, and 17

Regarding the present grouping, the Examiner relies on the following excerpt from Ranger to make a prior art showing of appellant's claimed "wherein the process is identified by inspecting ... a file signature associated with the process." (See Claim 5 et al.)

> "The content inspection mechanism analyzes decrypted content for such things as virus patterns, keywords, unknown program format, clearance labels or any other content based criteria." (col. 2, line 29)

The Examiner further argues that "Ranger teaches determining whether digital information is encrypted." Appellant respectfully disagrees with this assertion. The disclosure of an encryption determination in no way meets appellant's claimed file signature, let alone identifying a process accessing a file based on a file signature associated with the process.

Yet again, appellant respectfully asserts that at least the third element of the *prima facie* case of obviousness has not been met, since the prior art references, when combined, fail to teach or suggest all of the claim limitations, as noted above.

Group #5: Claims 24 and 25

Regarding the claims of the present grouping, such claims are deemed allowable for the reasons set forth hereinabove with respect to the first grouping of claims.

Further, it appears that the Examiner has overlooked numerous particular limitations in Claim 24 et al. Note, for example, the emphasized limitations noted below (which are non-existent in the remaining claims).

> "defining a plurality of extensions indicative of different types of files based on a user;
> identifying a file being accessed;
> determining the extension of the file being accessed; and
> performing virus detection actions on the file based on whether the extension is defined by the user;
> wherein at least a portion of the extensions relates to a plurality of processes each carried out by an executable file, the processes including at least one process initiated by an application program selected from the group consisting of a network browser application and a word processor application, for tailoring the virus detection actions when the application program attempts to access the file."

Again, appellant respectfully asserts that at least the third element of the *prima facie* case of obviousness has not been met, since the prior art references, when combined, fail to teach or suggest all of the claim limitations, as noted above.

Group #6: Claim 29

Regarding the claim of the present grouping, such claim is deemed allowable for the reasons set forth hereinabove with respect to the first grouping of claims. Still yet, it appears that the Examiner has overlooked numerous particular limitations in Claim 29. Note, for example, the emphasized limitations noted below (which are non-existent in the remaining claims).

> "(a) identifying a process for accessing files;

(b) selecting virus detection actions based at least in part on the process; and

(c) performing the virus detection actions on the files;

wherein the process is identified from a plurality of processes each carried out by an executable file, the processes initiated by application program-related executable files including FindFast.exe, WinWord.exe, and Explorer.exe, for tailoring the virus detection actions when attempts are made to access the files;

wherein the virus detection actions are selected by determining a category associated with the process, and selecting a set of virus detection actions based on the determined category;

wherein the process is identified by inspecting a name of the process, a path of the process, a file signature associated with the process, a version of the process, a manufacturer of the process, a function being called during the process, an owner of the process, a name of an executable file associated with the process, a method in which files are being accessed by the process, type(s) of shared libraries used by the identified process, and a user of the process" (**note the all-inclusive language**).

Again, appellant respectfully asserts that at least the third element of the *prima facie* case of obviousness has not been met, since the prior art references, when combined, fail to teach or suggest all of the claim limitations, as noted above.

In view of the remarks set forth hereinabove, all of the independent claims are deemed allowable, along with any claims depending therefrom.

VIII APPENDIX OF CLAIMS (37 C.F.R. § 41.37(c)(1)(viii))

The text of the claims involved in the appeal (along with associated status information) is set forth below:

1. (Previously Amended) A method for on-access computer virus scanning of files in an efficient manner, comprising:
(a) identifying a process for accessing files;
(b) selecting virus detection actions based at least in part on the process; and
(c) performing the virus detection actions on the files;
wherein the process is identified from a plurality of processes each carried out by an executable file, the processes including at least one process initiated by an application program selected from the group consisting of a network browser application and a word processor application, for tailoring the virus detection actions when the application program attempts to access the files.

2. (Cancelled)

3. (Original) The method as recited in claim 1, wherein the virus detection actions are selected by determining a category associated with the process, and selecting a set of virus detection actions based on the determined category.

4. (Original) The method as recited in claim 1, and further comprising the steps of identifying the files being accessed, and selecting the virus detection actions based at least in part on the identity of the files.

5. (Original) The method as recited in claim 1, wherein the process is identified by inspecting at least one of a name of the process, a path of the process, a

file signature associated with the process, a version of the process, a manufacturer of the process, a function being called during the process, an owner of the process, a name of an executable file associated with the process, a method in which files are being accessed by the process, type(s) of shared libraries used by the identified process, and a user of the process.

6. (Original) The method as recited in claim 1, wherein no virus detection actions are selected upon the identification of a predetermined process.

7. (Previously Amended) A computer program product embodied on a computer readable medium for on-access computer virus scanning of files in an efficient manner, comprising:

(a) computer code for identifying a process for accessing files;

(b) computer code for selecting virus detection actions based at least in part on the process; and

(c) computer code for performing the virus detection actions on the files;

wherein the process is identified from a plurality of processes each carried out by an executable file, the processes including at least one process initiated by an application program selected from the group consisting of a network browser application and a word processor application, for tailoring the virus detection actions when the application program attempts to access the files.

8. (Cancelled)

9. (Original) The computer program product as recited in claim 7, wherein the virus detection actions are selected by determining a category associated with the process, and selecting a set of virus detection actions based on the determined category.

10. (Original) The computer program product as recited in claim 7, and further comprising computer code for identifying the files being accessed, and selecting the virus detection actions based at least in part on the identity of the files.

11. (Original) The computer program product as recited in claim 7, wherein the process is identified by inspecting at least one of a name of the process, a path of the process, a file signature associated with the process, a version of the process, a manufacturer of the process, a function being called during the process, an owner of the process, a name of an executable file associated with the process, a method in which files are being accessed by the process, type(s) of shared libraries used by the process, and a user of the process.

12. (Original) The computer program product as recited in claim 7, wherein no virus detection actions are selected upon the identification of a predetermined process.

13. (Previously Amended) A system for on-access computer virus scanning of files in an efficient manner, comprising:

(a) logic for identifying a process for accessing files;

(b) logic for selecting virus detection actions based at least in part on the process; and

(c) logic for performing the virus detection actions on the files;

wherein the process is identified from a plurality of processes each carried out by an executable file, the processes including at least one process initiated by an application program selected from the group consisting of a network browser application and a word processor application, for tailoring the virus detection actions when the application program attempts to access the files.

14. (Cancelled)

15. (Original) The system as recited in claim 13, wherein the virus detection actions are selected by determining a category associated with the process, and selecting a set of virus detection actions based on the determined category.

16. (Original) The system as recited in claim 13, and further comprising logic for identifying the files being accessed, and selecting the virus detection actions based at least in part on the identity of the files.

17. (Original) The system as recited in claim 13, wherein the process is identified by inspecting at least one of a name of the process, a path of the process, a file signature associated with the process, a version of the process, a manufacturer of the process, a function being called during the process, an owner of the process, a name of an executable file associated with the process, a method in which files are being accessed by the process, type(s) of shared libraries used by the process, and a user of the process.

18. (Original) The system as recited in claim 13, wherein no virus detection actions are selected upon the identification of a predetermined process.

19. – 23. (Cancelled)

24. (Previously Amended) A method for computer virus scanning of files in an efficient manner, comprising:
defining a plurality of extensions indicative of different types of files based on a user;
identifying a file being accessed;
determining the extension of the file being accessed; and
performing virus detection actions on the file based on whether the extension is defined by the user;
wherein at least a portion of the extensions relates to a plurality of processes each carried out by an executable file, the processes including at least one

process initiated by an application program selected from the group consisting of a network browser application and a word processor application, for tailoring the virus detection actions when the application program attempts to access the file.

25. (Previously Amended) A computer program product embodied on a computer readable medium for computer virus scanning of files in an efficient manner, comprising:

computer code for defining a plurality of extensions indicative of different types of files based on a user;

computer code for identifying a file being accessed;

computer code for determining the extension of the file being accessed; and

computer code for performing virus detection actions on the file based on whether the extension is defined by the user;

wherein at least a portion of the extensions relates to a plurality of processes each carried out by an executable file, the processes including at least one process initiated by an application program selected from the group consisting of a network browser application and a word processor application, for tailoring the virus detection actions when the application program attempts to access the file.

26. (Previously Amended) A method for computer virus scanning of files in an efficient manner, comprising:

identifying a file being accessed;

identifying a process for accessing the file;

determining a category associated with the process;

selecting a set of virus detection actions based on the determined category;

determining an extension of the file being accessed; and

performing the virus detection actions on the files based on the extension wherein the process is identified from a plurality of processes each carried out by an executable file, the processes including at least one process initiated by an application program selected from the group consisting of a

network browser application and a word processor application, for tailoring the virus detection actions when the application program attempts to access the file.

27. (Previously Amended) A method for computer virus scanning of files in an efficient manner, comprising:
defining a plurality of extensions indicative of different types of files based on a user;
defining a plurality of categories indicative of different types of processes;
identifying a file being accessed;
identifying a process for accessing the file;
determining a category associated with the process;
selecting a set of virus detection actions based on the determined category;
determining the extension of the file being accessed; and
if the extension is defined by the user, performing the virus detection actions on the files;
wherein the process is identified from a plurality of processes each carried out by an executable file, the processes including at least one process initiated by an application program selected from the group consisting of a network browser application and a word processor application, for tailoring the virus detection actions when the application program attempts to access the files.

28. (Previously Amended) A method for on-access computer virus scanning of files in an efficient manner, comprising:
identifying a process for accessing files;
selecting virus detection actions based at least in part on the process; and
performing the virus detection actions on the files;
wherein downloading of infected files from the Internet is prevented;
wherein the process is identified from a plurality of processes each carried out by an executable file, the processes including at least one process

initiated by an application program selected from the group consisting of a network browser application and a word processor application, for tailoring the virus detection actions when the application program attempts to access the files.

29. (Previously Presented) A method for on-access computer virus scanning of files in an efficient manner, comprising:

(a) identifying a process for accessing files;

(b) selecting virus detection actions based at least in part on the process; and

(c) performing the virus detection actions on the files;

wherein the process is identified from a plurality of processes each carried out by an executable file, the processes initiated by application program-related executable files including FindFast.exe, WinWord.exe, and Explorer.exe, for tailoring the virus detection actions when attempts are made to access the files;

wherein the virus detection actions are selected by determining a category associated with the process, and selecting a set of virus detection actions based on the determined category;

wherein the process is identified by inspecting a name of the process, a path of the process, a file signature associated with the process, a version of the process, a manufacturer of the process, a function being called during the process, an owner of the process, a name of an executable file associated with the process, a method in which files are being accessed by the process, type(s) of shared libraries used by the identified process, and a user of the process.

IX APPENDIX LISTING ANY EVIDENCE RELIED ON BY THE APPELLANT IN THE APPEAL (37 C.F.R. § 41.37(c)(1)(ix))

There is no such evidence.

In the event a telephone conversation would expedite the prosecution of this application, the Examiner may reach the undersigned at (408) 971-2573. For payment of any additional fees due in connection with the filing of this paper, the Commissioner is authorized to charge such fees to Deposit Account No. 50-1351 (Order No. NAI1P004).

Respectfully submitted,

By: _____ Date: 4/1/05

Kevin J. Zilka
Reg. No. 41,429

Zilka-Kotab, P.C.
P.O. Box 721120
San Jose, California 95172-1120
Telephone: (408) 971-2573
Facsimile: (408) 971-4660